The Impact of Automated Grading System on Students' Assignments

Boaz Ben-Moshe[1] Nitza Davidovich[1]

[1]Ariel University Center of Samaria, Israel

The 6th Annual MEITAL National Conference: New Directions in E-Learning in Higher Education

Outline

AGS: a computerized tool to test programable assignments

A **complete system** which allows an **automated grading** for **programable** assignments

Preliminaries

- Computer program: text, formal language
- Compilation: translating a computer program to machine (runable) code.
- Running and testting: run the program and test it for several scenarios
- Grading process: performs tests and reports: errors, runtime, overall results.

Demonstration: C++ program, compilation, running.

Motivation for AGS

Current state

1. Most programs are tested manually (or not at all)
2. Time (money) consuming
3. limited testing, unfair grading.

Who needs AGS and why

1. Who: Students which study programming: Eng' CS, Phy...
2. Why: more reliable grading, saves money, simulate the 'real-world' (industry).
3. More reasons why: cheaters hate it, instant feedback...

General Structure of AGS.

Sub-systems

- A Publication sys': allows publishing assignments and grades
- The Submission sys': allows submitting assignments
- The Authentication sys': similarity tests (plagiarism detection tools)
- A Grading sys': tests and grades the assignments (the main module)

Lets demonstrate: assignments #1 in '201' class:

Implementing AGS

Implementation details

- We have implemented a **prototype** version of AGS, which allows grading programable assignments in C, C++, Java.
- The authentication sys is base on Stanford's MOSS[1] project.
- Other systems were implemented as web application.

Demonstrating

- Submission sys'
- Authentication tests (using MOSS)
- Auto-Grading output
- Related issues: web applications, 3 trails.

Experimental results with AGS

Experiment

- 400 students, from 10 classes, 3 departments.
- 3 lecturers, 4 letters of complains, 6 cheating cases.
- 1200 assignments were tested, 3 MS bugs, 1 year of work.

Results

- 21% less failures in '101' courses.
- Instant checking encourages student to resubmit improved version of their assignments..
- By product results: deeper understunding of program testing methodologies.

Disadvantages of AGS

Everybody hate the system!

- Students: hate the formality of the AGS, bad students hate the authentication sys.
- Lecturers: hate the need to define the assignments in fine details.
- Biro: hate the extra work implied by the AGS: complains, dealing with cheaters...

We are **strongly motivated** by the reasons the general public dislike the AGS!, yet:

- AGS: is complected (by nature)
- Platforms independent problems: Windows, Linux, Mac...
- Regulation are needed

AGS from the student point of view

Students don't like AGS

- Fast respond and several-try sys are 'nice to have'.
- Students tend to fear from AGS, and rather have their assignments graded manually.
- Self grading in small classes is not working

AGS from the lecturer point of view

Lecturers might like AGS

- More AG testing are needed, mainly in ’101’ courses.
- Using AGS requires a significant overhead in preparation (of the assignments)
- AGS can replace large portion of the manual grading - but NOT all of it.
- AGS should also be used as a QA testing

Summary

Main conclusion

- AGS should be friendly: MYCIN[2] case study.
- AGS should be used for large ’101’ classes, maybe even in High School
- AGS is a powerfull technical methodology which helps the average student.

FIN

Questions?

- Now
- By Email: benmo ariel.ac.il